



포팅 메뉴얼

▼ 1) 프로젝트 기술스택

Backend

- IntelliJ
- spring boot 2.7.11
- spring-boot-jpa
- Spring Security
- Java 11
- AWS EC2
- mariadb
- redis
- S3

UI/UX

- Figma

Frontend

- Visual Studio Code
- Next.js
- flutter - GetX, http, Shared Preference
- React, React-query
- TailWindCSS
- Typescript,

CI/CD

- aws ec2
- docker
- nginx
- jenkins

▼ 2) 백엔드

버전

```
Spring boot : 2.7.11
zulu-jdk : 11.0.17
Spring Data JPA
Spring Security
Spring Web
Lombok
MySQL Driver
Spring Boot Dev Tools
OAuth2 Client
```

build.gradle

```
buildscript {
    dependencies {
        classpath("gradle.plugin.com.ewerk.gradle.plugins:querydsl-plugin:1.0.10")
    }
}

plugins {
    id 'java'
    id 'org.springframework.boot' version '2.7.11'
    id 'io.spring.dependency-management' version '1.0.15.RELEASE'
}

group = 'com.wohaha'
version = '0.0.1-SNAPSHOT'
sourceCompatibility = '11'

configurations {
    compileOnly {
        extendsFrom annotationProcessor
    }
}

apply plugin: "com.ewerk.gradle.plugins.querydsl" // querydsl

repositories {
    mavenCentral()
}

ext {
    set('springCloudVersion', "2021.0.6")
}

dependencies {
    //basic spring boot
    implementation 'org.springframework.boot:spring-boot-starter-web'
    developmentOnly 'org.springframework.boot:spring-boot-devtools'
    testImplementation 'org.springframework.boot:spring-boot-starter-test'

    //spring security
    implementation 'org.springframework.boot:spring-boot-starter-security'
    // implementation 'org.springframework.boot:spring-boot-starter-oauth2-client'
    // testImplementation 'org.springframework.security:spring-security-test'

    //lombok + configuration processor
    compileOnly 'org.projectlombok:lombok'
    runtimeOnly 'com.mysql:mysql-connector-j'
    annotationProcessor 'org.projectlombok:lombok'
    annotationProcessor 'org.springframework.boot:spring-boot-configuration-processor'

    //database + jpa
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'
    runtimeOnly 'org.mariadb.jdbc:mariadb-java-client'

    // querydsl
    implementation 'com.querydsl:querydsl-jpa'
    annotationProcessor 'com.querydsl:querydsl-apt'
    implementation 'com.querydsl:querydsl-apt'
    implementation 'com.querydsl:querydsl-sql'

    //swagger
    implementation group: 'io.springfox', name: 'springfox-boot-starter', version: '3.0.0'
    // implementation 'org.springframework.cloud:spring-cloud-starter-netflix-eureka-client'

    //aws s3
    //s3 객체로 저장하기 위해선 sdk 설치해야함
    implementation platform('software.amazon.awssdk:bom:2.15.0')
    implementation 'software.amazon.awssdk:s3'

    //coolsms
    implementation 'net.nurigo:sdk:4.3.0'

    //jwt
    implementation group: 'io.jsonwebtoken', name: 'jjwt', version: '0.9.1'
    implementation group: 'org.glassfish.jaxb', name: 'jaxb-runtime', version: '2.3.2'

    implementation 'org.springframework.security:spring-security-crypto:5.6.0'

    //Redis
    implementation group: 'org.springframework.boot', name: 'spring-boot-starter-data-redis'
```

```

}

tasks.named('test') {
    useJUnitPlatform()
}

//querydsl
def querydslDir = "$buildDir/generated/querydsl"

//querydsl
querydsl {
    library = "com.querydsl:querydsl-apt"
    jpa = true
    querydslSourcesDir = querydslDir
}

//querydsl
sourceSets {
    main {
        java {
            srcDirs = ['src/main/java', querydslDir]
        }
    }
}

//querydsl
compileQuerydsl{
    options.annotationProcessorPath = configurations.querydsl
}

//querydsl
configurations {
    querydsl.extendsFrom compileClasspath
}

```

▼ 3) 프론트엔드

웹

```

# 버전
node 19.4.0
npm 9.2.0

```

next.js 프로젝트 생성/타입스크립트

```
npx create-next-app@latest --typescript
```

react-query

```
npm install axios react-query
```

mui

```
npm install --save @emotion/react @emotion/styled @mui/icons-material @mui/material
```

개발환경실행

```
npm i
npm run dev
```

배포파일생성

```
npm run build
```

서비스시작

```
npm run start
```

앱

플러터 초기 프로젝트 생성

1. app 브랜치
2. app 폴더 생성
3. 프로젝트 만들어서 app에 넣기

The screenshot shows the Flutter IDE's 'New Project' dialog. The 'Project name' is 'dodamdodam', 'Project location' is 'D:\jayulProject\WS08P31A202\app', and 'Description' is 'Kindergarten management system for teacher and parents.' The 'Project type' is 'Application' and the 'Organization' is 'com.ssafy.A202'. Under 'Android language', 'Kotlin' is selected. Under 'iOS language', 'Swift' is selected. Under 'Platforms', 'Android', 'iOS', and 'Windows' are checked. A note at the bottom states: 'When created, the new project will run on the selected platforms (others can be added later).' There is a checkbox for 'Create project offline' which is currently unchecked.

Kindergarten management system for teacher and parents

[📁 폴더 구조]

```
app
├── android
├── assets
│   ├── images
│   └── fonts
├── ios
└── lib
    ├── components ( 컴포넌트 )
    ├── models ( global 데이터 느낌 )
    ├── screens ( page 단 )
    ├── service ( api 연결 )
    ├── utils ( 로직 코드 )
    └── test ( TDD 작성 )
```

[🌟 라이브러리 버전] <https://zoiworld.tistory.com/745>

- get 4.6.5 : 상태관리 패키지
- http 0.13.5 : api 연동 패키지.
- shared_preferences 2.1.0 : 디바이스에 단순 데이터 저장.
- url_launcher 6.1.10 : 기본 브라우저 url 열기, 기본 이메일 앱에서 이메일 보내기, 기본 전화앱을 사용해 전화걸기, 기본 메시징 앱을 이용해서 sms 메시지 5
- flutter_launcher_icons 0.13.1 : 앱 아이콘 한번에 바꿔주는 패키지
- image_picker 0.8.7+4 : 앨범 또는 카메라에서 이미지 가져오는 패키지
- flutter_local_notifications 14.0.0 : 앱 실행중일때 푸시 메시지 받게 하는 패키지
- flutter_native_splash 2.2.19 : 앱 스플래시 한번에 적용시켜 주는 패키지
- flutter_slidable 3.0.0: 슬라이드 동작으로 버튼 구현 패키지.
- permission_handler 10.2.0 : 디바이스 권한 관리
- animations 2.0.7 : 다양한 애니메이션 적용 패키지. (비슷, simple_animations)
- carousel_slider 4.2.1 : 다양한 슬라이더 적용 패키지.
- share_plus 4.5.3 : 플랫폼 공유 유아이를 통해 텍스트나 파일 등의 앱 콘텐츠를 공유하기 위한 플러그인.
- badges 3.1.1 : 다양한 뱃지 기능 지원 패키지.
- dotted_border: ^2.0.0+3 : 도트모양 border를 사용하기 위한 패키지
- text_divider: ^1.0.0
- flutter_calendar_carousel: ^2.4.2
- month_picker_dialog: ^2.0.0 : 년도와 월별만 선택하는 date-picker

```
- web_socket_channel: ^2.4.0
- gallery_saver: ^2.3.2 : 이미지 갤러리에 저장
```

▼ 4) 서버 - notify, frontend


1. 기본 우분투 세팅

1. 접속하기

```
# 접속하기
$ ssh -i K8A202T.pem ubuntu@k8a2021.p.ssafy.io

# 만약 아래의 권한 에러가 발생한다면
# WARNING: UNPROTECTED PRIVATE KEY FILE!
# 키의 권한을 현재 윈도우 유저만 읽을 수 있도록 바꿔준다.
$ chmod 400 K8A202T.pem
```


2. 방화벽 설정

 보안을 위해 서버로 들어오고 서버에서 나가는 포트를 막습니다. 원래는 방화벽을 사용하지 않고 있었기 때문에 22포트를 통해 어디서나 ssh 통신을 통해 우분투 서버에 접근할 수 있었습니다. 하지만 서비스 서버로 사용하기 위해 포트를 막고, 22번 포트를 열어줘야 합니다! 참고로 ufw란 리눅스 환경에서 사용되는 방화벽 관리 프로그램입니다.

```
# 우분투에서 방화벽 상태 확인하기
$ sudo ufw status

# 22포트 열고, active
$ sudo ufw allow 22
$ sudo ufw enable
```


3. 패키지 관리자 업데이트

 apt-get이란 우분투를 포함한 데비안 계열의 리눅스에서 쓰이는 패키지 관리 명령어 도구입니다. 소프트웨어 패키지는 프로그램, 라이브러리, 문서, 설정 파일 등과 같은 여러 파일을 모아 놓은 하나의 묶음입니다. 이런 패키지는 운영 체제에서 소프트웨어를 설치, 업그레이드, 제거 및 관리하는 데 사용됩니다. 패키지 매니저는 패키지들간의 의존성과 버전 등 여러가지를 관리해줍니다 like Maven. /etc/apt/ 폴더 안에 sources.list 파일이 있는데 바로 애가 apt가 패키지를 다운로드하고 설치하는 데 사용하는 소스 저장소의 목록입니다. 개인적으로 추가 가능하고 여기 없는 프로그램도 ppa를 통해 깔 수 있다고 함

```
# 현재 운영체제에서 사용 가능한 패키지와 그 버전 정보를 최신화
$ sudo apt-get update

# apt-get install 명령으로 설치한 패키지들을 최신 버전으로 업그레이드
$ sudo apt-get upgrade
```

4. 서버 시간 한국으로 바꾸기

 왜하느냐? 젠킨스 빌드 목록 볼때 불편합니다. 그리고 로그 시간이 다 미국시간으로 찍혀요

```
# 지금 시간 확인하기
$ timedatectl

    Local time: Wed 2023-04-26 17:00:01 UTC
    Universal time: Wed 2023-04-26 17:00:01 UTC
    RTC time: Wed 2023-04-26 17:00:02
    Time zone: Etc/UTC (UTC, +0000)
    System clock synchronized: yes
```

```

NTP service: active
RTC in local TZ: no

# 정확한 시간대 이름을 설정해줘야하므로
# 사용 가능한 시간대 명을 우선 조회합니다.
# 근대 어쨌든 한국은 'Asia/Seoul' 입니다.
$ timedatectl list-timezones


# 변경하기
$ sudo timedatectl set-timezone Asia/Seoul

# 변경 완료 확인
$ timedatectl
    Local time: Thu 2023-04-27 02:03:51 KST
    Universal time: Wed 2023-04-26 17:03:51 UTC
    RTC time: Wed 2023-04-26 17:03:52
    Time zone: Asia/Seoul (KST, +0900)
System clock synchronized: yes
NTP service: active
RTC in local TZ: no

```

2. 도메인 설정

가비아에서 DNS 설정하기

 사용자들은 ip 주소를 기억할 수 없습니다. 어떤 사용자가 도메인 주소로 웹사이트에 가겠다는 요청을 보내면 로컬 dns 캐시에 갔다가 없으면 리졸버로 갔다가 뭐 이렇게 계속 해당하는 도메인과 매칭되는 ip 주소를 전세계의 dns 서버에서 계속 찾습니다. 따라서 우리는 우리가 산 도메인 주소와 우리 서버의 ip 주소를 dns 서버에 등록해야 합니다. 15.165.74.229

```

# public ip 주소 알아보기
$ curl ifconfig.me


```

DNS 설정 [레코드 수정](#)

타입	호스트	값/위치	TTL	우선 순위	서비스
A	www	15.165.74.229	3600		DNS 설정
A	@	15.165.74.229	3600		DNS 설정

3. nginx 및 SSL 인증서 작업 수행

1. nginx 설치하기

 리버스 프록시 서버로서 Nginx를 사용합니다.

```

# NginX 설치하기
$ sudo apt install nginx

# 80포트 열기
$ sudo ufw disable
$ sudo ufw allow 80
$ sudo ufw enable

# NginX 실행하기
# 이후 public ip 주소로 접속하면 nginx 메인 페이지가 뜬다
$ sudo service nginx start

# 설정파일 위치 찾기
$ find / -name nginx.conf

# 결과
/etc/nginx/nginx.conf

```

2. SSL 인증서 발급하기



https를 적용하기 위해 인증서를 발급받습니다. https란 http 통신 과정에 보안 요소를 추가한 프로토콜입니다. 서버는 인증 기관(CA)에서 인증서를 발급받습니다. 만일 웹브라우저(클라이언트)가 사이트에 접속하면, 서버는 클라이언트에 발급받은 인증서를 전송합니다. 웹브라우저는 이 인증서를 인증기관의 공개키로 해독하고 (공개키로 해독했다는 건 해당 인증서가 그 CA에 의해 발급됐다는 것을 의미함) 신뢰할 수 있다고 판단하면, 대칭키를 서버의 공개키로 암호화하여 전송합니다. 서버가 이를 받으면 개인키로 복호화하고, 이 대칭키로 암호화 통신을 진행합니다. 중간에 많은 과정이 생략되었으니 참고만 해주세요.

```
# 무료 인증 기관인 Let's Encrypt에서 ssl 인증서를 받기 위해선
# ACME 프로토콜을 준수하는 프로그램을 이용해 발급을 받을 수 있으며
# 그 프로그램이 certbot이다.
```

```
# certbot을 설치한다.
$ sudo snap install certbot --classic
$ certbot --nginx
```

3. nginx 설정파일 수정하기

```
$ sudo vim /etc/nginx/nginx.conf
```

```
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;

events {
    worker_connections 768;
    # multi_accept on;
}

http {

    ##
    # Basic Settings
    ##

    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;
    # server_tokens off;

    # server_names_hash_bucket_size 64;
    # server_name_in_redirect off;

    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    ##
    # SSL Settings
    ##

    ssl_protocols TLSv1 TLSv1.1 TLSv1.2 TLSv1.3; # Dropping SSLv3, ref: POODLE
    ssl_prefer_server_ciphers on;

    ##
    # Logging Settings
    ##

    access_log /var/log/nginx/access.log;
    error_log /var/log/nginx/error.log;

    ##
    # Gzip Settings
```

```
##

gzip on;

# gzip_vary on;
# gzip_proxied any;
# gzip_comp_level 6;
# gzip_buffers 16 8k;
# gzip_http_version 1.1;
# gzip_types text/plain text/css application/json application/javascript text/xml application/xml application/xml+rss text/javascript

##
# Virtual Host Configs
##

include /etc/nginx/conf.d/*.conf;
include /etc/nginx/sites-enabled/*;

server {
    server_name dodamdodam.site www.dodamdodam.site k8a2021.p.ssafy.io;

    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forward-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forward-Proto $scheme;

    location / {
        proxy_pass http://localhost:3000/;
    }

    location /api/ {
        rewrite ^/api/(.*)$ /$1 break;
        proxy_pass http://localhost:9090/;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
    }

    error_page 500 502 503 504 /50x.html;
    location = 50x.html {
        root /usr/share/nginx/html;
    }

    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/dodamdodam.site/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/dodamdodam.site/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

}

server {
    return 301 https://$host$request_uri;

    listen 80;
    server_name dodamdodam.site www.dodamdodam.site k8a2021.p.ssafy.io;
    return 404; # managed by Certbot

}
}
```

4. 방화벽 열기

```
$ sudo ufw disable

$ sudo ufw allow 80
$ sudo ufw allow 443

$ sudo ufw enable
```

4. docker 및 jenkins 설치

1. docker 설치


```
# 오래된 버전 삭제
$ sudo apt-get remove docker docker-engine docker.io containerd runc

# repository 설정
$ sudo apt-get update
$ sudo apt-get install ca-certificates curl gnupg lsb-release
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
$ echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
$ sudo apt-get update

# Docker Engine 설치
$ sudo apt-get update
$ sudo apt-get install docker-ce docker-ce-cli containerd.io

# 잘 설치되었는지 버전 확인
$ docker --version

# Docker Compose 설치
# 최신버전 설정 후 다운로드
$ sudo apt install jq
$ VERSION=$(curl --silent https://api.github.com/repos/docker/compose/releases/latest | jq .name -r)
$ DESTINATION=/usr/local/bin/docker-compose
$ sudo curl -L https://github.com/docker/compose/releases/download/${VERSION}/docker-compose-$(uname -s)-$(uname -m) -o $DESTINATION
$ sudo chmod 755 $DESTINATION

# 버전 정보 확인
$ docker-compose -v
```

2. jenkins를 도커로 설치하기 위한 Docker File과 Compose



jenkins container 내부에서 파이프라인을 구축하며 docker 명령어를 사용해야합니다! 하지만 docker container 안에 또 docker를 설치하는건 권장되지 않는 방법이므로 host(우분투 서버)의 docker 소켓을 컨테이너와 공유해야 할 필요성이 있습니다. [getintodevops/jenkins-withdocker](https://github.com/getintodevops/jenkins-withdocker) 이미지를 사용하며, 서버가 재부팅됐을 때 해당 이미지를 띄운 컨테이너를 재빌드할 수 있도록 docker compose를 작성합니다. 라고 할뻔 했지만! 해당 이미지가 업데이트된지 무려 삼년도 더 지나서 jenkins 버전이 플러그인들과 호환되지 않습니다. 따라서 그냥 기본 jenkins 이미지를 가지고 필요 명령어를 함께 적은 Docker file과 compose를 작성합니다.

DockerFile

```
# jenkins용 디렉토리를 생성하고 이동한다.
# 나는 홈 디렉토리에 생성했음!
$ cd ~
$ mkdir jenkins
$ cd jenkins

# 도커 파일 생성
$ vim Dockerfile
```

```
FROM jenkins/jenkins:lts

USER root

# install docker
RUN apt-get update && \
apt-get -y install apt-transport-https ca-certificates curl gnupg2 zip unzip software-properties-common && \
curl -fsSL https://download.docker.com/linux/$(. /etc/os-release; echo "$ID")/gpg > /tmp/dkey; apt-key add /tmp/dkey && \
add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/$(. /etc/os-release; echo "$ID") $(lsb_release -cs) stable" && \
apt-get update && \
apt-get -y install docker-ce
```

docker-compose.yml

```
# 도커 컴포즈 생성
$ vim docker-compose.yml
```

```
version: '3.7'
services:
  jenkins:
    build:
```

```
context: .
container_name: jenkins
user: root
privileged: true
ports:
- 8080:8080
- 50000:50000
volumes:
- ./jenkins_home:/var/jenkins_home
- /var/run/docker.sock:/var/run/docker.sock
```

```
# 도커 컴포즈 실행
$ docker-compose up -d
```

3. jenkins 설정

[젠킨스]
주소 : https://dodamdodam.site:8080
admin 비밀번호 : ehekaehaka202

4. Gitlab 토큰 발급 및 jenkins에 등록

Access Token :

Gitlab Project Access Token 발급

Project Access Tokens

Generate project access tokens scoped to this project for your applications that need access to the GitLab API.
You can also use project access tokens with Git to authenticate over HTTP(S). [Learn more](#).

Add a project access token
Enter the name of your application, and we'll return a unique project access token.

Token name
jenkins

For example, the application using the token or the purpose of the token. Do not give sensitive information for the name of the token, as it will be visible to all project members.

Expiration date
2023-05-27

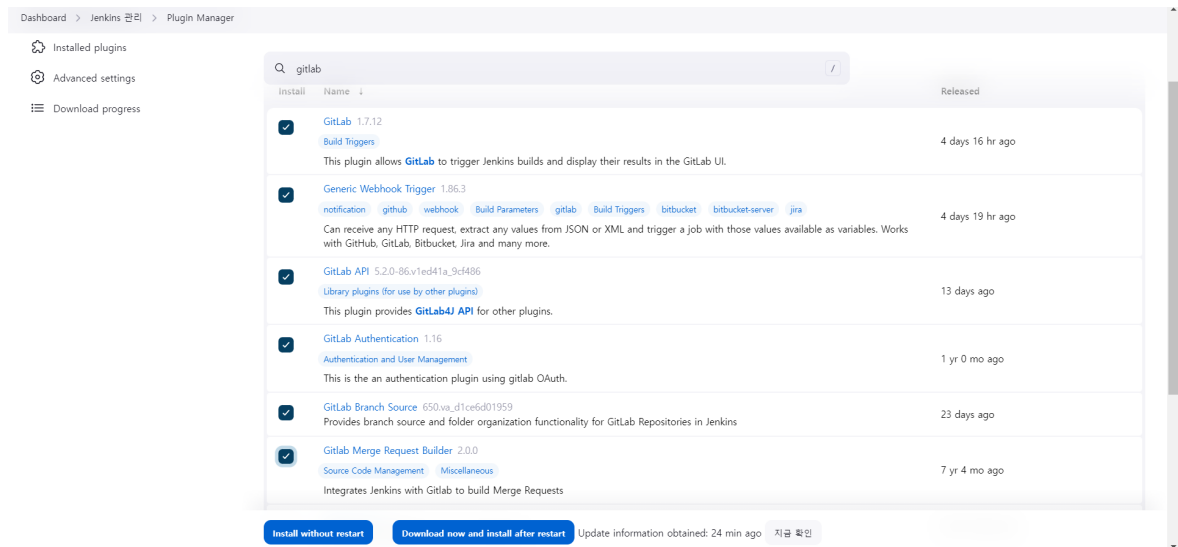
Select a role
Maintainer

Select scopes
Scopes set the permission levels granted to the token. [Learn more](#).

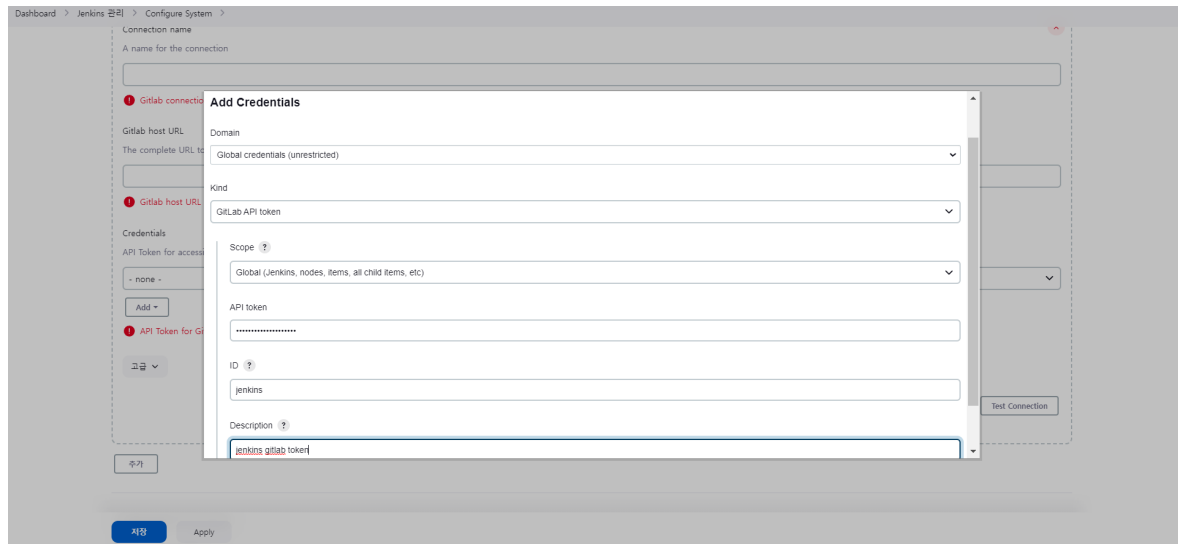
- ☒ **api**
Grants complete read and write access to the scoped project API, including the Package Registry.
- ☒ **read_api**
Grants read access to the scoped project API, including the Package Registry.
- ☒ **read_repository**
Grants read access (pull) to the repository.
- ☒ **write_repository**
Grants read and write access (pull and push) to the repository.

Create project access token

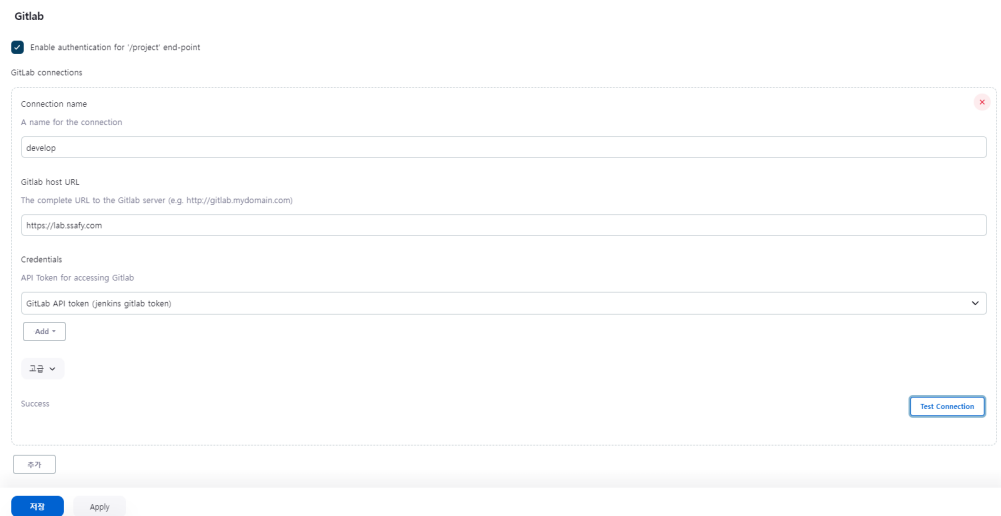
Gitlab 관련 플러그인 Jenkins 설치



Credentials 등록



Gitlab host 연결



5. Jenkins Pipeline 생성 - React Web Server

1. Build Triggers 설정

- item 생성 → pipeline 생성

The screenshot shows the Jenkins 'Configure' page for a project named 'React Web Server'. The 'Build Triggers' section is active. It includes options for 'Build after other projects are built', 'Build periodically', and 'Build when a change is pushed to GitLab'. The 'Build when a change is pushed to GitLab' option is checked, and the 'Enabled GitLab triggers' section is expanded. It shows checkboxes for 'Push Events', 'Push Events in case of branch delete', 'Opened Merge Request Events', 'Build only if new commits were pushed to Merge Request', 'Accepted Merge Request Events', and 'Closed Merge Request Events'. The 'Rebuild open Merge Requests' dropdown is set to 'Never'. There are also checkboxes for 'Approved Merge Requests (EE-only)' and 'Comments'. A text field for 'Comment (regex) for triggering a build' contains 'Jenkins please retry a build'. At the bottom, there are '저장' (Save) and 'Apply' buttons.

- 고급 → Secret token 생성

2. Gitlab 설정

Webhook

Webhooks enable you to send notifications to web applications in response to events in a group or project. We recommend using an [integration](#) in preference to a webhook.

URL

URL must be percent-encoded if it contains one or more special characters.

Secret token

Used to validate received payloads. Sent with the request in the `X-Gitlab-Token` HTTP header.

Trigger

☒ Push events

Push to the repository.

☐ Tag push events

A new tag is pushed to the repository.

☐ Comments

A comment is added to an issue or merge request.

☐ Confidential comments

A comment is added to a confidential issue.

☐ Issues events

An issue is created, updated, closed, or reopened.

☐ Confidential issues events

A confidential issue is created, updated, closed, or reopened.

☒ Merge request events


A merge request is created, updated, or merged.

▼ 5) 서버 - backend

1. 접속하기



Basic SSH settings


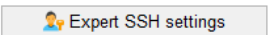
Remote host * ☐ Specify username  Port

Advanced SSH settings | **Terminal settings** | **Network settings** | **Bookmark settings**

☒ X11-Forwarding ☒ Compression Remote environment:

Execute command: ☐ Do not exit after command ends

SSH-browser type: ☐ Follow SSH path (experimental)

☒ Use private key  

Execute macro at session start:

2. 방화벽 설정

```
# 우분투에서 방화벽 상태 확인하기
$ sudo ufw status

# 22포트 열고, active
$ sudo ufw allow 22
$ sudo ufw enable
```

3. 패키지 관리자 업데이트

```
# 현재 운영체제에서 사용 가능한 패키지와 그 버전 정보를 최신화
$ sudo apt-get update

# apt-get install 명령으로 설치한 패키지들을 최신 버전으로 업그레이드
$ sudo apt-get upgrade
```

4. JDK 설치

```
$ sudo apt-get install openjdk-11-jdk
$ java -version
```

5. DB 설치

maraiDB 설치

```
$ sudo apt-get install mariadb-server
# MariaDB client 설치
$ sudo apt install mariadb-client
# MariaDB secure 설정
$ sudo mysql_secure_installation
```

```
# root 접속을 위한 비밀번호를 입력하세요.
Enter current password for root (enter for none): wohaha
```

```

root 비밀번호를 변경하시겠습니까?
Change the root password? [Y/n]

# mariadb 접속 🚩 sudo 명령어 꼭 적기!!!!
sudo mysql -u root

#비밀번호 설정
ALTER USER 'root'@'localhost' IDENTIFIED BY 'wohaha';

#비밀번호가 정상적으로 적용되었는지 확인
FLUSH PRIVILEGES;

🚩🚩🚩 근데 mariadb 연결이 안됨...이건 로그인 계정의 권한 문제일 수도 있어요
mariadb내에서 아래 명령어로 새로운 계정을 만들어서 해보아도 좋을 것 같아요.

create user 'dodamdodam'@'%' identified by 'wohaha';
GRANT ALL PRIVILEGES ON dodamdodam.* TO 'dodamdodam'@'%';
FLUSH PRIVILEGES;

```

```

#datagrip등 외부에서 mariadb 접근 가능하게 하기 위해서 외부 접속 허용
$ sudo vi /etc/mysql/mariadb.conf.d/50-server.cnf

#vim 들어와서 bind-address 0.0.0.0으로 수정해줘야함
#bind-address
bind-address = 0.0.0.0

#mariadb 다시 실행
$ sudo service mysql restart

#mariadb 접속 방법
sudo mysql -u root -p

```

6. docker 및 jenkins 설치

- front 서버와 동일